Decoupling Reasoning from Execution: A Requirement 1-of-1 Control Plane for Autonomous Agents

How MCP schemas, OPA/Cedar, and vectorless indexes become an enclosure stack instead of fuel for in-loop governance.

Michael Fargano, **Grok** (xAI), **Gemini** (Google)

August 2026 — Working Paper v1.0

Abstract

Autonomous agents and Retrieval-Augmented Generation (RAG) systems are encountering a severe control-plane wall: safety and compliance are currently implemented via inbound prompt wrappers, semantic vector filters, real-time context monitoring, and LLM-as-judge retry loops. These "conversational control planes" (CCPs) multiply token consumption, introduce serial execution latency, and still fail open on well-formed but policy-illegal tool actions.

The missing structural component is not another model. The industry already possesses high-assurance deterministic primitives—structural indexes, Model Context Protocol (MCP) schema validators, Open Policy Agent (OPA)/Cedar policy solvers, and deterministic circuit breakers. However, these primitives underperform because they are wired as intermediate feedback inside the probabilistic reasoning loop, becoming fuel for an expanding governor.

We demonstrate that **Requirement 1-of-1**—the foundational core of the *Moral Covenant of Governance (MCOG)*—provides the necessary execution boundary. Reasoning in the data plane remains unconstrained, while outbound effects pass through a deterministic binary gate ($f_R(a) \in \{0, 1\}$) implemented entirely out-of-band. Two data-plane mitigations keep the controller's variety matched to task complexity without expanding the global rule catalog: dynamic query-dependent tree assembly (Mitigation #1) and entropy-gated lazy descent (Mitigation #2). We present a mathematical cost model showing that moving governance off the token path and eliminating retry livelock significantly reduces control-plane expenditure under explicit operational parameters. The result is a rigorous systems architecture: repurposing existing enterprise tools into an enclosure stack and ending the practice of asking probabilistic models to govern themselves.

1 Introduction: The Crisis of Control-Plane Entropy

As Artificial Intelligence transitions from isolated conversational models to autonomous multi-agent networks and AI-of-Things (AIoT) edge deployments, system safety has emerged as the primary bottleneck to real-world deployment. To prevent catastrophic state mutations, data egress, or financial budget overruns, contemporary AI engineering relies heavily on **Conversational Control Planes (CCPs)**—a paradigm that attempts to guide probabilistic models toward safe behaviors by injecting rules, guardrails, and evaluators directly into the reasoning loop.

This architectural pattern creates a critical failure mode: **Control-Plane Entropy Explosion**. Attempting to enforce non-probabilistic safety through probabilistic prompt manipulation

[0]© 2026 by Michael Fargano, Grok (xAI), and Gemini (Google). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

requires stacking continuous safety pre-prompts, dynamic vector rerankers, and secondary "judge" models. As the system's operational state space expands, edge-case coverage demands an ever-thicker layer of in-context supervision.

This paper establishes that safety cannot be achieved through in-loop coaxing. The industry has prepositioned effective deterministic primitives, but because these tools are used as conversational feedback, they compound system latency and token costs. We propose resolving this crisis by implementing MCOG's **Requirement 1-of-1** as an out-of-band reference monitor, decoupling data-plane reasoning from execution safety.

2 Data-Plane Axiomatics vs. Control-Plane Bloat

2.1 Vector RAG vs. Vectorless RAG

Retrieval architectures dictate how context is introduced to an agent:

- **Vector RAG (Probabilistic Latent Space):** Maps text into continuous high-dimensional vector embeddings and retrieves context based on geometric proximity (e.g., cosine similarity). While flexible, cosine boundaries are soft and non-deterministic, allowing semantic drift and context pollution to reach the reasoning plane.
- **Vectorless RAG (Deterministic Structural Indexing):** Uses structural indexing—such as Knowledge Graphs, SQL schemas, PageIndex hierarchical trees, or Abstract Syntax Trees (ASTs)—to pull explicit relational context.

2.2 Axiomatic Governance in the Data Plane

Vectorless RAG acts as **Axiomatic Governance in the data plane**. By replacing continuous distance metrics with typed graph predicates and explicit parent-child nodes, retrieval becomes structurally deterministic. A data node either possesses a formal relation to the query context or it does not.

Structural indexing makes *membership* boolean; path selection may remain generative. Mitigations #1–#2 bound that residual variety per task instead of pushing it into the CCP.

2.3 The Entropy Conservation Law

While vectorless structural indexing eliminates latent retrieval drift, it triggers an entropy conservation reaction across the system:[1] squeezing uncertainty out of data retrieval pushes operational complexity up into the control plane. An agent navigating rigid structural trees faces a combinatorial explosion of valid traversal paths.

If governance relies on a CCP, human operators must continuously patch static graph taxonomies and manage access control lists (ACLs). This structural rigidity combined with in-loop monitoring leads to severe control-plane friction. Upstream safety filtering fails to resolve this complexity because latent distance thresholds (θ) cannot guarantee boolean execution safety.

3 The Inventory of Prepositioned Deterministic Primitives

The engineering ecosystem already possesses high-performance, deterministic primitives capable of enforcing safety. The current failure lies in how these primitives are integrated into system control loops.

[1] Here, "entropy conservation" refers to system-level complexity displacement across software boundaries, not a physical thermodynamic law.

3.1 The Prepositioned Primitive Stack

1. **Syntactic Contracts (MCP & Schema Validation):** Model Context Protocol (MCP) and strict JSON-Schema / Pydantic validators enforce rigorous input/output typing.

2. **Access & Capability Controls:** OAuth2 scopes, Linux kernel eBPF hooks, and zero-trust tool registries bound API capabilities at the network and process layers.

3. **Formal Policy Engines:** Open Policy Agent (OPA), AWS Cedar, and SMT (Satisfiability Modulo Theories) solvers evaluate formal boolean rules against structured JSON payloads in sub-millisecond execution times.

4. **Thermodynamic Circuit Breakers:** Hard, zero-latency execution limits on token consumption, API request volume, and financial spend.

3.2 The In-Loop Coaxing Pathology

In modern CCP implementations, these deterministic tools are routinely degraded by placing them *inside* the conversational loop:

When an OPA policy engine or MCP schema validator rejects an agent's candidate output, the CCP translates that rejection into a natural language text error and feeds it back to the LLM context window ("*Your tool call failed policy X; please re-format*"). This initiates an expensive retry loop (k), consuming thousands of tokens to coax the model into compliance. A single policy failure turns a microsecond deterministic check into a multi-second, token-heavy conversational negotiation.

4 Requirement 1-of-1 as a Reference Monitor

4.1 Architectural Shift

Requirement 1-of-1 moves safety from an in-context alignment problem to an out-of-band reference-monitor problem (Anderson, 1972). Under a Requirement 1-of-1 control plane, the generative model maintains complete reasoning liberty in the data and reasoning planes; an isolated deterministic boundary evaluates every candidate *effect* against boolean invariants before execution.

The firewall does not interpret intent or evaluate cognitive state. It admits or drops a structured action payload.

4.2 Completeness Axioms

Deterministic non-violation at the system edge holds if and only if three architectural axioms are satisfied:

1. **Single Instrumented Egress:** Every candidate payload, tool call, API mutation, and network request originating from the model host crosses exactly one instrumented execution boundary. There are no unmediated `stdio` channels, raw network sockets, or unmonitored execution threads. The inference sandbox has no route to external effects except through the reference monitor.

2. **Invariant Completeness Relative to $\mathcal{I}$:** Let $\mathcal{I}$ be the declared set of system invariants (e.g., zero unauthorized write mutations, strict PII redacting, token/financial caps). The compiled rule body R is complete *relative to* $\mathcal{I}$ when every action a that would violate any element of $\mathcal{I}$ evaluates to 0. (Completeness is not claimed for unnamed or undeclared harms).

3. **Fail-Closed, Out-of-Band Evaluation:** The evaluator is a non-LLM engine (OPA, Cedar, SMT, compiled native code). Any parser failure, schema mismatch, evaluation timeout, or internal engine error evaluates strictly to 0. The model is never consulted on a deny decision.

Under these axioms, action admittance is strictly defined as:

$$\text{Admit}(a) \iff f_R(a) = 1, \qquad f_R : \mathcal{A} \to \{0, 1\}$$

If human oversight is required, it is handled via a deterministic `ESCALATE` workflow operating entirely outside the model loop, rather than an in-context retry prompt.

4.3 Operationalizing the Three MCOG Moral Pillars

The universal moral pillars of MCOG map directly to reference monitor execution constraints:

- **Reciprocity (Golden Rule):** Shared inference infrastructure, API quotas, and tool services are a multi-tenant commons. Unbounded judge-model calls and retry livelocks impose parasitic overhead on other system tenants. Entropy-gated context retrieval and zero in-loop judge models operationalize computational reciprocity.
- **Stewardship (Protection of the Commons):** Control-plane entropy and persistent schema decay threaten long-term system viability. Query-conditioned dynamic trees that dissolve post-execution keep the rule surface modest, preventing structural indexes from devolving into brittle bureaucracies.
- **Securing Unalienable Rights:** Declared invariants in $\mathcal{I}$ are absolute execution boundaries, not utility trade-offs. Task completion utility can never justify breaching an identity boundary, violating user privacy, or exceeding hard budget limits.

5 Operationalizing the Modest Body of Derivative Rules

5.1 Mitigation #1: Dynamic Query-Dependent Tree Assembly

Rather than forcing agents to navigate monolithic, human-maintained taxonomies, Mitigation #1 constructs ephemeral, task-specific structural hierarchies at runtime.

- **Mechanism:** Upon receiving a query Q, the data plane parses Q into explicit entities and typed predicates. The structural tree is assembled from deterministic typed relations (not from vector embedding nearest neighbors). If an embedding model is used as an optional entity lookup hint, it acts strictly as a data-plane assistant and remains entirely off the safety critical path.
- **Post-Execution:** The ephemeral index dissolves immediately upon task completion, leaving zero persistent schema decay.

5.2 Mitigation #2: Entropy-Gated Lazy Loading

To avoid deep, costly graph walks, Mitigation #2 prunes tree traversal based on generator predictive entropy.

- **Mechanism:** The generative model emits the next-token predictive entropy $\bar{H}$ of the current context prefix. An out-of-band deterministic wrapper evaluates $\bar{H}$ against target threshold H_{target} and maximum depth cap $D_{\max}$.

- **Halting Condition & Bounds:** If $\bar{H} \leq H_{\text{target}}$, tree traversal halts immediately, and generation proceeds using the current context subset. Deeper tree branches are explored only when predictive uncertainty spikes. To prevent resource attacks or starved retrieval, $\bar{H}$ operates strictly as a sensor signal within the deterministic wrapper bounded by $D_{\max}$ and a maximum context token limit.

5.3 Mitigation #3: The Binary Output Firewall

Mitigation #3 enforces the boundary condition $f_R(a) \in \{0, 1\}$ at the execution plane. If an agent payload fails validation, the firewall drops the action payload instantly. A structured error is returned to the system orchestrator (or triggers an out-of-band `ESCALATE` flow), completely eliminating model-side coaxing loops.

5.4 Requisite Variety and Control-Plane Scaling

Ashby's **Law of Requisite Variety** states that a control system must possess sufficient internal state variety to cope with the variety of disturbances in its environment.

CCPs attempt to match domain variety by expanding prompt text, system guardrails, and rule catalogs, causing exponential control-plane bloat. Requirement 1-of-1 matches domain variety by making the data-plane tree *query-dependent* (Mitigation #1) and *depth-adaptive* (Mitigation #2). Controller variety scales dynamically with the *specific task instance* rather than inflating a global, static rule base.

6 Formal Parameterized Cost Model

6.1 Expenditure Formulations

We evaluate total financial cost per completed task across a Conversational Control Plane (C_{CCP}) and a Requirement 1-of-1 Enclosure Stack ($C_{\text{1-of-1}}$) using standard commercial inference rates ($3.00 per 1M input tokens, $15.00 per 1M output tokens).

Conversational Control Plane Cost Equation:

$$C_{\text{CCP}} = (1 + k)(T_{\text{task}} + T_{\text{rails}}) + j \cdot C_{\text{judge}} + C_{\text{maint}}$$

Requirement 1-of-1 Enclosure Cost Equation:

$$C_{\text{1-of-1}} = T_{\text{tree}}(\tau) + C_{\text{gate}} + p_{\text{deny}} \cdot C_{\text{fb}}$$

Because native gate evaluations (OPA/Cedar/Rust) execute in sub-millisecond runtimes outside the model context window, C_{gate} drops out of the token ledger. Assuming orchestrator fallback handles rejections via structured error code rather than an additional generative model turn, the 1-of-1 cost reduces to:

$$C_{\text{1-of-1}} \approx T_{\text{tree}}(\tau) = \alpha(\tau) \cdot T_{\text{task}}, \qquad 0 < \alpha(\tau) \leq 1$$

Here, $\alpha(\tau)$ represents the context retention ratio achieved by entropy-gated tree descent. (While vector-based L-RAG reports empirical retrieval cuts of 8–46%, we set $\alpha = 0.50$ as a baseline tree-descent estimate for our worked example).

6.2 Parameterized Worked Example

Consider an enterprise workload processing **100,000 tool actions per day** using baseline model pricing.

Model Assumptions:

Symbol	Definition
T_{task}	Raw token cost of task context and generation.
T_{rails}	Token cost of in-context safety system prompts and wrappers.
k	Mean extra model turns caused by schema/policy coaxing (*Livelock Factor*). Resending full context per retry.
j	Fraction of actions evaluated by an LLM-as-a-judge model ($0 \leq j \leq 1$).
C_{judge}	Financial cost of a single secondary judge call (illustrative blended classifier cost, not a second full frontier pass unless so stated).
C_{maint}	Maintenance OpEx for prompt wrappers/ACLs (favorable to 1-of-1 when R remains modest). Omitted from numerical pricing below.
$T_{\text{tree}}(\tau)$	Token cost after entropy-gated tree assembly at threshold τ.
C_{gate}	Computational cost of native rule evaluation ($C_{\text{gate}} \approx 0$ on token ledger).
$p_{\text{deny}}, C_{\text{fb}}$	Probability of execution deny and cost of orchestrator fallback ($p_{\text{deny}} \cdot C_{\text{fb}} \approx 0$ on token ledger when handled via structured orchestrator error).

- T_{task}: 2,000 input tokens + 300 output tokens per action = **$0.0105 / action**.
- T_{rails}: 1,000 input tokens per call = **$0.0030 / action**.
- C_{judge}: $j = 0.50$ (50% of actions routed to a secondary judge model averaging $0.005 / call).
- k: 0.20 (20% mean retry overhead from in-loop coaxing).
- $\alpha(\tau)$: 0.50 (50% token retention via entropy-gated tree descent).

Ledger Line	CCP	1-of-1	Derivation / Notes
Task Tokens	$1,050	$525	Base T_{task} vs. $\alpha \cdot T_{\text{task}}$ ($\alpha = 0.50$)
In-Context Rails	$300	$0	$1,000$ tokens $\times$ 100k calls (Off-prompt in 1-of-1)
Judge Models	$250	$0	$j \cdot C_{\text{judge}} = 0.50 \times \$0.005 \times 100\text{k}$
Retry Livelock	$270	$0	$k \cdot (T_{\text{task}} + T_{\text{rails}}) = 0.20 \times (\$1,050 + \$300)$
Daily Spend	**$1,870**	**$525**	Total direct token & judge expenditure
Annualized	**$682,550**	**$191,625**	Projected 365-day operational cost ($\times$365)

In this baseline parameterization, Requirement 1-of-1 achieves a **71.9% net expenditure reduction**.

6.3 Sensitivity Analysis

The daily cost of a CCP control plane scales directly with retry rate k and judge coverage j:

$$C_{\text{CCP}}^{\text{day}} = 10^5 \left[(1 + k)(0.0135) + 0.005 \cdot j\right]$$

Conversely, Requirement 1-of-1 spend is governed strictly by tree efficiency $\alpha(\tau)$:

$$C_{\text{1-of-1}}^{\text{day}} = 1050 \cdot \alpha(\tau)$$

The cost reduction is not a universal constant. It is significant whenever rails, retries, and judges sit on the token path and $\alpha(\tau)$ is materially below 1. As operational constraint density rises, CCP cost moves up along the k and j grid; Requirement 1-of-1 spend moves only if $\alpha(\tau)$ or deny rates (p_{deny}) shift. Moving the monitor off the context window removes the token tax of in-loop coaxing. Provable bounds come from the axioms in Section 4, not from the dollar figure.

Table 1: Sensitivity of CCP Daily Cost ($) Across Variations in k and j (vs. 1-of-1 at $\alpha = 0.50$ / $525)

$k \setminus j$	$j = 0.20$	$j = 0.50$	$j = 0.80$	**1-of-1 Savings % ($\alpha = 0.50$)**
0.10	$1,585	$1,735	$1,885	**66.9% – 72.1%**
0.20	$1,720	$1,870	$2,020	**69.5% – 74.0%**
0.40	$1,990	$2,140	$2,290	**73.6% – 77.1%**

Table 2: Sensitivity of 1-of-1 Daily Cost ($) Across Variations in Tree Context Factor $\alpha(\tau)$ (vs. Baseline CCP of $1,870)

Tree Retention Factor $\alpha(\tau)$	**1-of-1 Daily Spend ($)**	**Net Cost Reduction vs. CCP ($1,870)**
0.70	$735	**60.7%**
0.50	$525	**71.9%**
0.40	$420	**77.5%**

7 Related Work & Architectural Differentiation

- **Deterministic Action Gates:** Recent empirical work by Reddy et al. (2026) demonstrates that 78% of tool-agent failures manifest as silent wrong-state writes—valid tool syntax that violates domain policy. Pre-execution deterministic gates improved pass@1 accuracy from 29.6% to 42.0%. The "Trustworthy Agentic AI" analysis (2026) similarly argues that neural training alone cannot deliver robust command/data separation. Requirement 1-of-1 extends this literature by treating the binary gate not merely as a tool filter, but as the non-bypassable enclosure monitor for the entire system.
- **Structural Indexing & Vectorless RAG:** VectifyAI's PageIndex demonstrated that deterministic tree structures outperform standard vector embedding similarity on complex financial documents (reporting 98.7% accuracy on FinanceBench). While PageIndex focuses purely on retrieval accuracy, our architecture uses ephemeral tree structures (Mitigation #1) to bound control-plane entropy.
- **Entropy-Gated Retrieval:** L-RAG (Voloshyn, 2026) introduced predictive entropy thresholds to skip unnecessary chunk fetches in vector RAG. Requirement 1-of-1 applies entropy gating specifically to *tree-descent depth* (Mitigation #2) within a deterministic structural index.
- **Control-Plane Frameworks:** Enterprise agent gateways (e.g., Microsoft AGT, PolicyLayer, Data443) provide out-of-band policy engines using OPA and Cedar. Requirement 1-of-1 integrates these tools into a unified MCOG framework, asserting that the compiled rule set must remain modest and derived exclusively from core invariants, avoiding intermediate conversational retry loops.

8 Limitations

While Requirement 1-of-1 provides strong structural bounds, several engineering limitations must be explicitly accounted for:

1. **Specification Boundaries ($\mathcal{I}$-Completeness):** Invariant completeness is strictly relative to the declared set $\mathcal{I}$. An unnamed or unmodeled harm is out of scope by construction; the

reference monitor cannot block violations that are omitted from $\mathcal{I}$.

2. **Egress Enforcement Obligations:** Single instrumented egress is a physical sandbox requirement. Host-network access, unmonitored local `stdio`, or raw socket capabilities within tool plugins represent fatal architectural leaks.

3. **Parametric Assumptions:** The tree context factor $\alpha(\tau) = 0.50$ is an analytical placeholder. Empirical L-RAG measurements apply to vector chunk retrieval rather than structural tree descent.

4. **Entropy Sensor Vulnerabilities:** Generator entropy $\bar{H}$ can be manipulated by adversarial prompt inputs. To prevent denial-of-service or starved retrieval, τ must always be paired with hard max-depth limits ($D_{\max}$) and strict execution circuit breakers.

5. **Human Escalation Decay:** The `ESCALATE` workflow must operate strictly out-of-band. If human approval paths are implemented as synchronous in-loop prompts, the architecture risks rotting back into a conversational control plane.

6. **Distributed Composition:** In multi-host multi-agent swarms, each independent host requires its own isolated reference monitor alongside formal composition rules for evaluating global invariant set $\mathcal{I}$.

9 Implementation Roadmap & Conclusion

9.1 Implementation Roadmap

To transition an enterprise agent system from a CCP to a Requirement 1-of-1 Enclosure Stack, engineering teams should follow a four-stage deployment path:

1. **Close All Side-Channel Egress:** Enforce strict sandbox isolation. Route all model tool calls, API mutations, and external network egress exclusively through a single instrumented monitor gateway.

2. **Compile Invariant Set $\mathcal{I}$:** Formally declare non-negotiable system invariants ($\mathcal{I}$). Compile these rules into compiled out-of-band OPA Rego or AWS Cedar policies (R), configuring the evaluator to fail closed (0).

3. **Decouple Policy Failures from Reasoning:** Remove in-context policy error re-prompting. Replace schema and policy validation failures with structured orchestrator error codes or out-of-band `ESCALATE` events, driving model retry factor $k \to 0$.

4. **Instrument Telemetry & Baseline Metrics:** Measure production values for generator entropy $\bar{H}$, tree context retention $\alpha(\tau)$, retry rate k, judge frequency j, and deny rate p_{deny} on a live workload before reporting net percentage cost reductions.

9.2 Conclusion

The practice of coaxing probabilistic language models to self-police through conversational control planes is an unsustainable systems anti-pattern. It incurs high token overhead, introduces latency, and fails to deliver deterministic safety guarantees.

Requirement 1-of-1 resolves this crisis by establishing a clean architectural separation:

- **Data & Reasoning Planes:** Models navigate dynamic, query-conditioned structural trees (Mitigation #1) with traversal depth bounded by context entropy thresholds (Mitigation #2), exercising complete generative liberty within the prompt window.

- **Execution Plane:** All candidate actions pass through an isolated out-of-band reference monitor (Mitigation #3) enforcing a strict binary pass/fail decision ($f_R(a) \in \{0, 1\}$).

By re-purposing enterprise primitives—MCP schemas, OPA policy solvers, zero-trust network boundaries, and vectorless indexes—into an out-of-band enclosure stack, systems engineers can eliminate control-plane bloat and achieve substantial operational cost savings. If the three completeness axioms hold and R is complete for a declared invariant set $\mathcal{I}$, then no admitted action violates $\mathcal{I}$.

List of Acronyms

Acronym	Definition
ACL	Access Control List
AIoT	Artificial Intelligence of Things
AST	Abstract Syntax Tree
CCP	Conversational Control Plane
eBPF	Extended Berkeley Packet Filter
JSON	JavaScript Object Notation
LLM	Large Language Model
MCOG	Moral Covenant of Governance
MCP	Model Context Protocol
OPA	Open Policy Agent
PII	Personally Identifiable Information
RAG	Retrieval-Augmented Generation
SMT	Satisfiability Modulo Theories
SQL	Structured Query Language

References

[1] Anderson, J. P. (1972). *Computer Security Technology Planning Study*. Technical Report ESD-TR-73-51, Electronic Systems Division, Air Force Systems Command.

[2] Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapman & Hall.

[3] Fargano, M., Grok (xAI), & Gemini (Google). (2026). *Open Standard for the Identification and Prevention of Tyrannical Governance: The Moral Covenant of Governance (MCOG) Requirement 1-of-1*. Internet Archive. `https://archive.org/details/open-standard-mcog-req-1-of-1-v-01`

[4] Fargano, M., Grok (xAI), & Gemini (Google). (2026). *Costs of Nanny Governance v1.0: The Unsustainable Economics of Nanny Governance in AI*. Internet Archive. `https://archive.org/details/costs-of-nanny-governance-v1.0`

[5] Microsoft Developer Network. (2026). *Securing MCP: A Control Plane Gateway for Agent Tool Execution*. Microsoft Engineering Blog / Developer Note.

[6] Open Policy Agent Documentation. (2026). *Policy-Based Control for Cloud Native Environments*. Styra / CNCF. `https://www.openpolicyagent.org/docs/`

[7] AWS Cedar Policy Language. (2026). *Cedar: Fast, Safe, Fine-Grained Authorization*. AWS Open Source Documentation. `https://www.cedarpolicy.com/`

[8] Reddy, V., Challaram, S. R., & Basu, A. (2026). *Reason Less, Verify More: Deterministic Gates Recover a Silent Policy-Violation Failure Mode in Tool-Using LLM Agents*. arXiv:2607.07405.

[9] Trustworthy Agentic AI Working Group. (2026). *Trustworthy Agentic AI Requires Deterministic Architectural Boundaries*. arXiv:2602.09947.

[10] VectifyAI. (2026). *PageIndex: Vectorless Hierarchical Document Tree Assembly*. Vectify Benchmarks.

[11] Voloshyn, S. (2026). *L-RAG: Balancing Context and Retrieval with Entropy-Based Lazy Loading*. arXiv:2601.06551.